



clases

atributos

relación

## Práctica 2.

invariantes

Queremos modelar un sistema que tiene coches de una marca concreta (por ejemplo, TOYOTA) que pueden viajar entre ciudades y se someten a revisiones, de acuerdo con las siguientes especificaciones.

1. El sistema registra **ciudades**, de modo que a cada par de ciudades las separan unos kilómetros concretos, que son los recorridos que harán los coches de nuestro sistema, y cada ciudad debe estar al menos a 5 kilómetros de distancia de otra.
2. Los **coches** tienen una **fecha de matriculación** y **registran los kilómetros recorridos**. También tienen información sobre si **están en garantía** y **necesitan mantenimiento**, como se explica posteriormente.
3. Los coches realizan **viajes**, de modo que cada **viaje tiene una fecha de salida y una fecha de llegada**. En un viaje, **el coche siempre realiza un recorrido entre dos ciudades**.
4. Los coches deben someterse a **revisiones**, donde toda revisión debe **tener una fecha de inicio y una fecha de fin**. Una revisión puede tratarse de un **trabajo de mantenimiento** o bien una **reparación de alguna avería**. **Todas las revisiones deben tener lugar después de que el coche se matricule.**
5. Como mucho, **un coche se debe someter a una revisión, como máximo, en un momento dado.**
6. Las revisiones se realizan en **talleres**, que pueden ser **oficiales** o **no**. Un **taller oficial** ofrece una **garantía de un tiempo determinado a toda revisión que se realice en su taller**. Cada **taller se sitúa en una ciudad**, y en **cada ciudad habrá, a lo sumo, un taller oficial**, pudiendo haber varios talleres no oficiales.
7. **Si un coche está siendo sometido a una revisión, entonces el coche debe encontrarse en la misma ciudad donde está el taller.**
8. En relación con los viajes que realiza un coche, todo viaje terminado tendrá una fecha de salida y una fecha de llegada. **Si el coche está realizando actualmente algún viaje, dicho viaje únicamente tendrá fecha de salida, pero no de llegada.**
9. **Un coche se encontrará en todo momento bien realizando un viaje determinado o bien en una ciudad.**
10. **Si el coche ha completado al menos un viaje y no se encuentra viajando, entonces debe encontrarse en la ciudad a la que llegó en su último viaje.**
11. **Dos viajes no pueden solaparse en el tiempo, es decir, un viaje debe ocurrir siempre después de otro, pudiendo la fecha de llegada de un viaje coincidir con la fecha de salida del siguiente.**
12. Las **ciudades de origen y destino** de los viajes deben ser coherentes. **Es decir, si un coche realiza un viaje desde la ciudad A hasta la ciudad B, el próximo viaje debe partir desde la ciudad B.**
13. **Los kilómetros que tiene un coche deben ser la suma de los kilómetros de los viajes que dicho coche ha completado.** En nuestro sistema, no nos debemos preocupar de insertar manualmente el número de kilómetros de un coche, pues debería estar actualizado en todo momento.

}DERIVE



Grado en Ingeniería del Software  
Modelado y Diseño de Software  
Curso 2024-2025

---

14. En este sistema, **todas las fechas serán un valor entero que representa un día.** **Vamos a suponer que tanto los viajes como las revisiones comienzan un día y terminan X días posteriores.** Asimismo, por simplicidad, vamos a suponer que un año son 100 días.
15. En el punto 2 se mencionó que los coches tienen información sobre si están en garantía. **Un coche está en garantía si no han pasado 4 años desde que se matriculó. También está en garantía si han pasado menos días que los indicados en la garantía del taller oficial donde el coche tuvo alguna revisión.**
16. En el punto 2 se mencionó que los coches tienen información sobre si necesitan mantenimiento. **Durante los primeros cuatro años desde la fecha de matriculación, un coche no necesita mantenimiento. Después de ese tiempo, un coche no necesita mantenimiento hasta pasado un año de la última revisión de mantenimiento que tuviera, sin importar el taller donde se realizó.**

En el entregable se distinguirán los siguientes apartados:

- (a) Desarrollar en UML, utilizando las herramientas USE y VisualParadigm, un modelo conceptual de la estructura de dicho sistema, con los correspondientes elementos, las relaciones entre ellos y todas las restricciones de integridad necesarias (esto último solamente en la herramienta USE). En lo relacionado con el tiempo, en este apartado podemos considerar que tenemos una clase Clock de la cual habrá una única instancia. Dicha clase tiene un atributo “now” que nos indica el día en que nos encontramos actualmente. No hay que modelar el paso del tiempo en este apartado. Se debe entregar la imagen del diagrama de clases y todo el código USE desarrollado. Si durante el desarrollo se ha creado un modelo conceptual en SOIL para comprobar el funcionamiento del sistema, se puede entregar también el código SOIL correspondiente (y cualquier imagen del modelo conceptual necesaria para explicarlo).
- (b) Este apartado consiste en modelar el comportamiento del sistema. En particular, se va a modelar únicamente el comportamiento de los coches cuando viajan. Para ello, vamos a registrar la velocidad de los coches. La velocidad de un coche indica cuántos kilómetros avanza cada día en los viajes. El sistema también va a almacenar los kilómetros de los viajes. Cuando un viaje está en curso, este atributo indica los kilómetros que lleva realizado. Cuando un viaje se completa, este atributo debe tener el mismo valor que los kilómetros del recorrido que ha realizado. Se deben modelar las siguientes acciones:
  - (b1) Un coche comienza un viaje desde la ciudad en la que se encuentra. Esta operación debe recibir como parámetro el recorrido entre dos ciudades que debe realizar en su viaje.
  - (b2) Una operación avanzar que se ejecuta sobre los coches, y que no recibe ningún parámetro. Esta operación debe hacer avanzar el coche el número de kilómetros indicados en su velocidad si el coche está realizando algún viaje.
  - (b3) Se debe modelar el paso del tiempo, de modo que un tic del reloj representa el paso de un día, lo cual se debe tener en cuenta a la hora de que los coches puedan avanzar en el viaje que estén realizando.



Grado en Ingeniería del Software  
Modelado y Diseño de Software  
Curso 2024-2025

---

En este apartado, se debe entregar la imagen del diagrama de clases y el código USE desarrollado (al ser este apartado un incremento respecto del anterior, hay que entregar únicamente el código USE nuevo). En las operaciones añadidas, especificar las pre- y post-condiciones.

- (c) Ahora se debe desarrollar un modelo de objetos y simularlo. Vamos a considerar tres ciudades: Málaga, Sevilla y Granada. Tendremos dos recorridos: entre Málaga y Sevilla con 210 kilómetros, y entre Sevilla y Granada con 250 kilómetros. Supondremos un coche matriculado en el instante 0 (día 0) y que viaja a una velocidad de 27. El coche comienza en Málaga y continúa en Málaga hasta el día 5, día en que comienza un viaje haciendo el recorrido de Málaga a Sevilla. Los días van pasando y el coche va avanzando hasta que llega a Sevilla. El mismo día que llega a Sevilla, el coche comienza otro viaje haciendo el recorrido entre Sevilla y Granada. Los días van pasando y el coche continúa realizando el viaje. Una vez llega a Granada, la simulación termina.

Se debe mostrar 3 imágenes del diagrama conceptual: una en el instante 0, otra cuando el coche llega a Sevilla y otra cuando el coche lleva a Granada. Entregar también el código SOIL necesario para reproducir el modelo conceptual y la simulación.

La práctica se entregará en dos archivos: uno comprimido que contendrá los proyectos Visual Paradigm y USE creados al efecto, y otro en formato PDF con una memoria que incluya \*todos\* los diagramas de clases y objetos desarrollados, las restricciones definidas, así como aquellas explicaciones textuales que puedan considerarse necesarias por no estar claras en los diagramas o que sean necesarias para facilitar la comprensión del modelo, sus entidades y relaciones.

El fichero comprimido debe denominarse con el nombre del grupo seguido del número de la práctica (GRX-X\_P2). Por ejemplo, el grupo GR1-1 deberá entregar un fichero denominado: GR1-1\_P2.zip. Igualmente, la memoria del grupo en formato PDF debe denominarse con el nombre del grupo seguido del número de la práctica (P1), por ejemplo GR1-1\_P2.pdf.

Se hará una sola entrega por grupo, por lo que solo será necesario que uno de los miembros del grupo de prácticas suba el archivo comprimido al campus virtual. No hay límite de páginas, aunque se ruega brevedad y concisión.

*El trabajo debe realizarse en grupo, por lo que cada grupo debe entregar la solución que haya modelado. La copia entre grupos será calificada negativamente. El uso de aplicaciones de IA generativa también será calificado negativamente. De hecho, durante el examen no se tendrá acceso a ninguna de estas aplicaciones, por lo que el estudiante debe tratar de aprender a modelar por sí mismo.*